Parallel Programming: Assignment 1

Notes: You may discuss the problems with your peers but the submitted work must be your own work. No late assignment will be accepted. Please submit your answers through blackboard. This assignment is worth 15 % of your total grade.

Important Notes:

- You may develop your programs locally on your computer but the performance datafor problem 3 and 4 should be collected on the virtual machine provided by ICT for you.
- Note that jobs may wait in the queue to be executed for couple hours on a busy day,thus plan accordingly and do not wait for the last day of the assignment.

Problem 1

(10 points) Problem 2.16 from the textbook.
a) Suppose the run-time of a serial program is given by $T_{serial} = n^2$, where the units of the run-time are in microseconds. Suppose that a parallelization of this program has run-time $T_{parallel} = n^2/p + log_2(p)$. Write a program that finds the speedups and efficiencies of this program for various values of n and p. Run your program with $n = 10, 20, 40, ..., 320$, and $p = 1, 2, 4, ..., 128$.
- What happens to the speedups and efficiencies as p is increased and n is held fixed?
- What happens when p is fixed and n is increased?

b) Suppose that $T_{parallel} = T_{serial}/p + T_{overhead}$ and also suppose that we fix p and increasethe problem size.
- Show that if $T_{overhead}$ grows more slowly than T_{serial}, the parallel efficiency will increase aswe increase the problem size.
- Show that if, on the other hand, $T_{overhead}$ grows faster than T_{serial}, the parallel efficiencywill decrease as we increase the problem size.

Problem 2

(10 points) In this problem, you are required to develop a performance model for the versionsof parallel sum in Lecture 4. Assume the followings
- Sequential execution time is S,
- Number of threads, T,
- Parallelisation overhead O (fixed for all versions),
- The cost B for the barrier or M for each innovation of the mutex
- Let N be the number of elements that we are summing.

(a) Using these variables, what is the execution time of valid parallel versions 2, 3 and 5(see Lecture 4). Note that for version 5, there is some additional work for thread 0 that you should also model using the variables above.
(b) Present a model of when parallelization is profitable for Version 3;

Problem 3

(10 points) Download the STREAM benchmark from http://www.cs.virginia.edu/stream/. Run it

on your computer or on a virtual machine provided for you by ICT and measure the performance of copy, scale, add and triad benchmarks usingthe same array size but varying number of OpenMP threads (1, 2, 4, 8, and 16). Report the benchmark results and explain your observations.

Problem 4

(70 points)

1. Write a program to calculate π in Pthreads for a number of iterations to be providedby the user. A serial implementation is provided on blackboard (pi-serial.c) that shouldbe used for validation and speedup.

   (a) What is the parallel speedup of your code? To compute parallel speedup, you will need to time the execution of both the sequential and parallel code (*speedup* = *Time*(*seq*)/*Time*(*parallel*)). Plot the speedup as a function of threads using 1, 2, 4, 8,16 threads on ICT provided VM for a fixed number of iterations. If your code does not speed up, you will need to adjust the parallelism granularity, the amount of work each processor does between synchronisation points. You can do this by increasing the number of iterations.

   (b) What is the parallel efficiency of your code? Plot the efficiency as a function ofthreads using 1, 2, 4, 8, 16 threads on the VM for a fixed number of iterations.

   (c) Report lines of code for the serial and Pthreads implementations.

2. Write the same computation in OpenMP and perform the studies in (a), (b), and (c). Submit

a report containing the plots along with your source codes.

Estimating π: We will be using the following approximation to estimate π :

$$\pi = 4[1 - 1/3 + 1/5 - 1/7 + ...] = 4 \sum_{k=0}^{\infty} \frac{(-1)^k}{2k+1}$$

Measuring time: You may use *gettimeofday()* from <sys/time.h> to record the start andend time of the computation.

Compiling and running your program: You may use GNU compiler for development on your local machine and the VM provide by ICT

For reliable performance measurements, your programs including the parallel versions shouldrun at least 5 seconds. Adjust the number of iterations as needed.